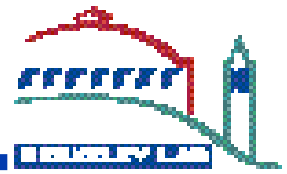


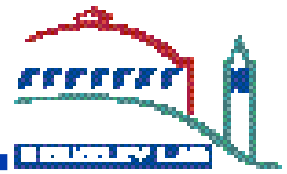
Cross-Platform Authentication With LDAP

Greg Haverkamp

ITSD UNIX Systems Group

March 17, 2004

Overview



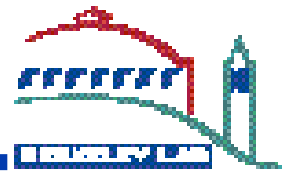
When assessing the situation at the Berkeley Center for Structural Biology (BCSB), we asked ourselves some questions:

- What problems are we trying solve?
- What are the goals we'd like to achieve?
- Why do we think LDAP will help?
- What is LDAP?

From there, we'll look at:

- Implementation overview
- Issues & Problems & Experiences
- Q&A

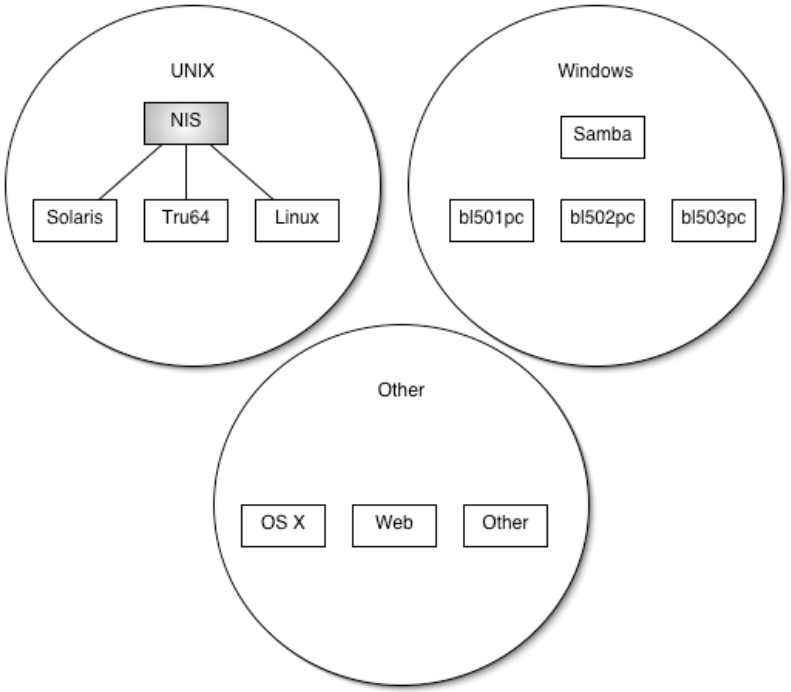
Problems to solve



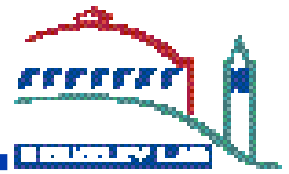
- Lots of outside users using facility
- Beamline support personnel needed some administrative control (add and modify users, change passwords)
- Group accounts
- Management of individual passwords
- End of published password lists
- Integration of key platforms, principally UNIX and Windows

Problems to solve

- Every platform an island
- Superuser privileges required to modify NIS passwords and add users (and beamline support personnel aren't sysadmins)
- No mechanism in place to keep Samba passwods in sync with NIS passwords
- Separate smbpasswd files on each Samba server

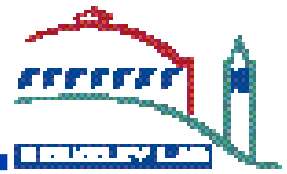


Goals



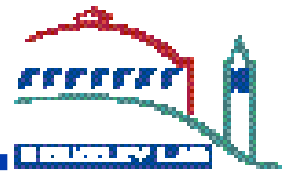
- Bring all user management into a single user database
- Delegate authority to beamline support personnel (SEAs)
- Make it easy for SEAs to modify users
- Eliminate group accounts
- Eventually, delegate some authority for users to their Principal Investigators (PIs)
- Provide readily accessible, replicated data store to allow authorized access to various machine classes

Why will LDAP help?



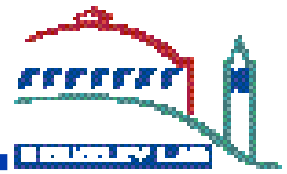
- Easy to distribute data
- Easy replication (especially with iPlanet/SunONE/Netscape Directory Server; but doable with OpenLDAP)
- Easy to delegate management
- Fine-grained access control (ACIs in iPlanet, ACLs in openLDAP)
- Combines multiple databases (passwd, shadow, group, smbpasswd, phone book, etc) in one database
- Operating system support
- Wide array of management utilities

Why will LDAP help? (cont'd)



- Easy to program in Perl (Net::LDAP) and Java (JNDI)
- Client authentication via password (binddn, proxy agent, etc.)
- Client authentication via client SSL certificate
- User authentication via SASL if desired (useful if you've got other uses for SASL)
- Communications can be encrypted via SSL (ldaps) or TLS

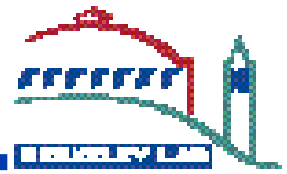
LDAP v. NIS



- NIS is tightly integrated with UNIX platforms
- NIS is simple and well-known
- NIS can be quick and easy to set up

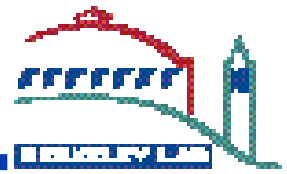
- NIS administration can't be easily delegated
- NIS is relatively insecure (passwords sent in the clear, world-readable)
- NIS stores a limited variety of information (passwd, group, netgroups, etc.)

What is LDAP?



- *LDAP (Lightweight Directory Access Protocol) provides a standards-based protocol used for accessing directory resources.*
- Often when people speak of LDAP they mean a directory server that implements an interface via LDAP. (I may fall prey to this.) I'll try to watch out for this.

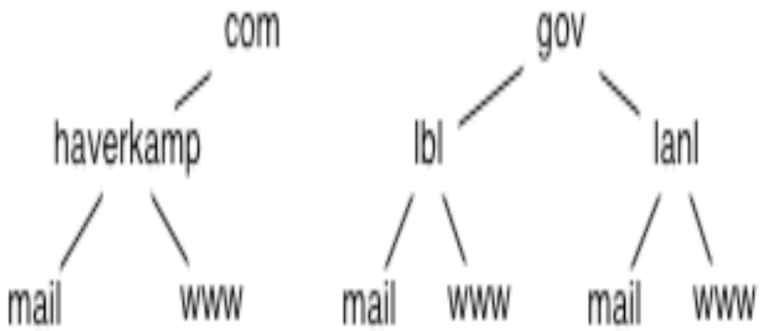
What's a directory?



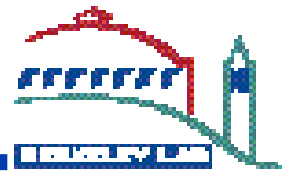
- *A directory is a data store consisting of (typically) hierarchically arranged data.*
- *A directory server is a specialized database optimized for searching and retrieving (typically) hierarchical data.*
- Popular directory servers include OpenLDAP, iPlanet Directory Server, MS Active Directory, and Novell's eDirectory

DNS

- DNS is probably the most widely used directory
- Useful analog because we understand it
- Moving down the tree, domain names formed are distinct throughout the system
- Note that storage backend doesn't matter for a directory; access method is all that really matters



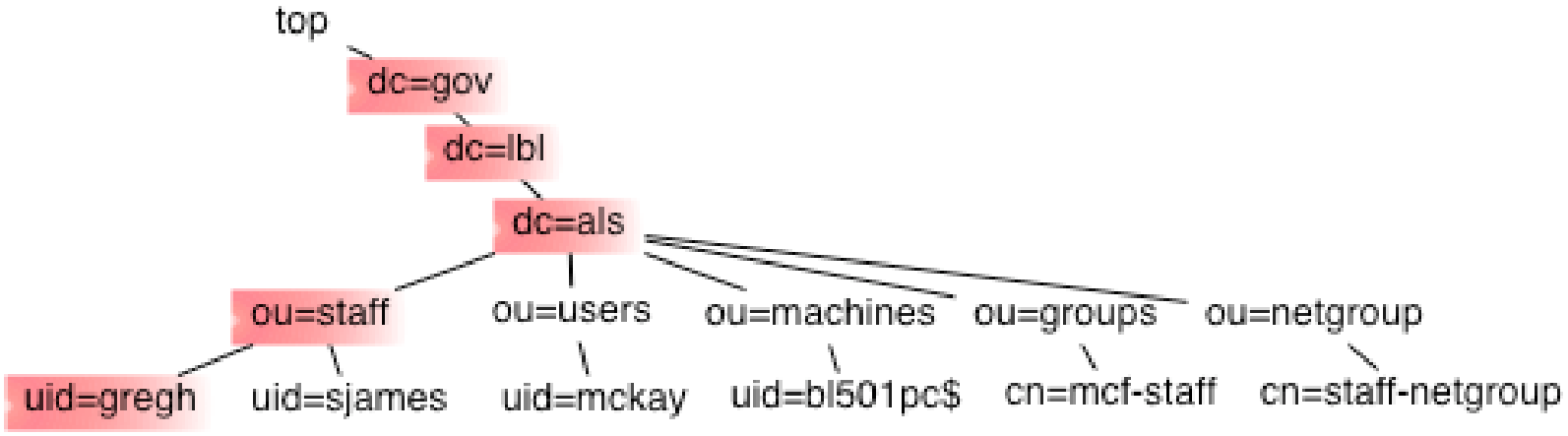
LDAP basics



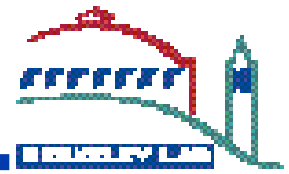
- Central to LDAP operations is the *Distinguished Name*.
- The DN is roughly equivalent to a FQDN.
- The DN represents a unique entry in a directory; no two entries can have the same DN (that's why it's *distinguished*.)
- DN makeup will vary by directory location, schema choices, and organizational implementation decisions.

Distinguished Names

- From RFC 2251: *The concatenation of the relative distinguished names of the sequence of entries from a particular entry to an immediate subordinate of the root of the tree forms that entry's Distinguished Name (DN), which is unique in the tree.*)



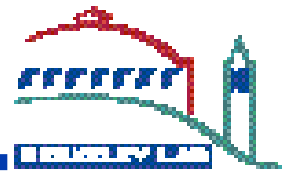
Attributes



- **Meat of directory entries come from *attributes*.**
- **Attributes are attached to DNs:**

```
dn:
uid=gregh,ou=staff,dc=als,dc=lbl,dc=gov
ntPassword: <password>
lmPassword: <password>
userPassword: <password>
homeDirectory: /home/staff/gregh
host: admin
host: staff
host: download
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowaccount
objectClass: sambaAccount
uidNumber: 30001
gidNumber: 5626
loginShell: /bin/tcsh
gecos: Greg Haverkamp
shadowLastChange: -1
shadowMin: -1
shadowMax: -1
shadowWarning: -1
shadowInactive: -1
shadowExpire: -1
shadowFlag: -1
uid: gregh
pwdLastSet: 1049493434
logonTime: 0
logofftime: 2147483647
kickoffTime: 2147483647
pwdCanChange: 2147483647
pwdMustChange: 2147483647
cn: gregh
description: Greg Haverkamp
smbHome: \\bcsb-smb\gregh
homeDrive: U:
scriptPath: startup.cmd
profilePath: \\bcsb-
smb\gregh\profiles
rid: 61002
primaryGroupID: 10535
acctFlags: [UX         ]
```

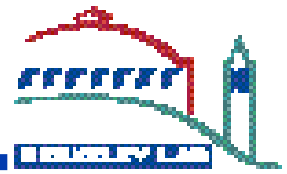
Groups representation



- Groups in Organization Unit “Groups”
- posixGroup gives the group the gidNumber, as well as optional password.
- groupOfNames allows group to use memberUid or uniqueMember attributes to define group members.

```
dn: cn=mcf-staff,ou=Groups,dc=als,dc=lbl,dc=gov
objectClass: posixGroup
objectClass: top
objectClass: groupOfNames
gidNumber: 5626
cn: mcf-staff
memberUid: cwcork
memberUid: earnest
memberUid: gerry
memberUid: sjames
memberUid: timossi
memberUid: ewcornel
memberUid: arthur
memberUid: mcfoper
```

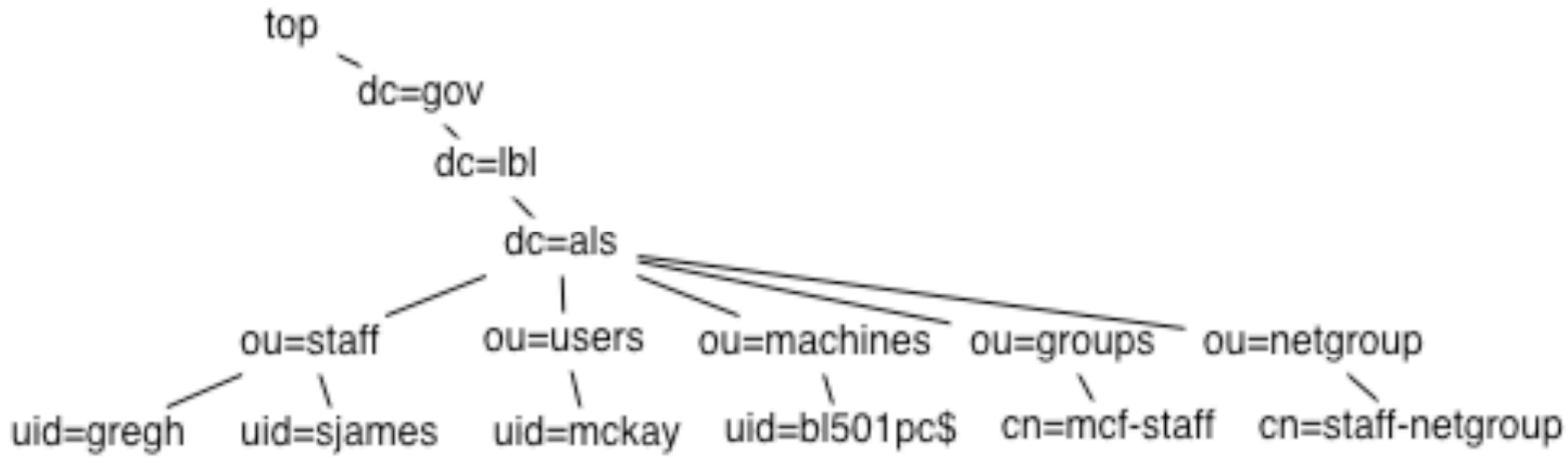
Implementation Overview



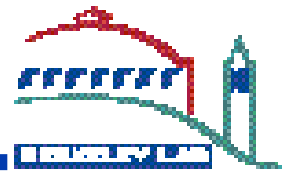
- BCSB directory implementation
- Implementation on UNIX
- Samba Implementation
- Apache authentication
- Tomcat authentication

BCSB Directory Hierarchy

- **Split into organizational units to more clearly separate classes of users.**

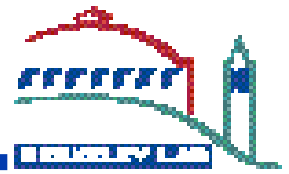


iPlanet v. OpenLDAP



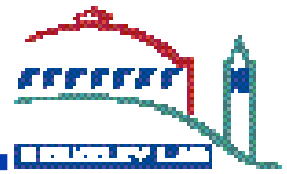
- **Several reasons we chose iPlanet Directory Server over OpenLDAP for this implementation:**
  - **iPlanet is tightly integrated with Solaris 9, and in fact is installed by default with Solaris 9**
  - **iPlanet out-of-the-box allows use of Solaris profiles, and Solaris is the predominant operating system in play.**
  - **OpenLDAP has deprecated LDAPv2 support, and until recently, it appeared Tru64 would be around for the long haul.**
  - **iPlanet supports multi-master replication, which better fits the BCSB model of separate-but-equal across sectors.**

Failover



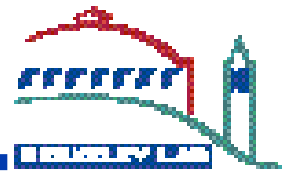
- Tru64 doesn't support failover between servers at all
- Linux failover, while advertised, has yet to function correctly for us in testing. It's possible this is still just an implementation issue.
- OS X failover, which should work similar to Linux failover, has not been tested (it's in very low use.)
- Solaris failover has been tested, and failover happens almost immediately. Recovery returns to default server.
- In general, I believe an application-level load balancer would do a better job of failover than the client libraries, when available.

UNIX Implementation



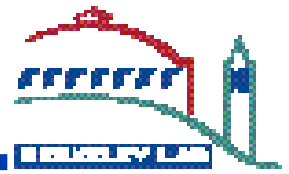
- Solaris
- Linux
- OS X
- Tru64

Implementation on Solaris



- **Solaris 8 and Solaris 9 ship with LDAP capabilities integrated**
- **ldap_cachemgr works in concert with nscd to cache data from the directory**
- **With iPlanet (and modifications to OpenLDAP) Solaris can use profiles to automatically update its profile from the LDAP server:**
- ***ldapclient -v init -a profileName=default -a proxyDN='cn=proxyagent,ou=profile,dc=als,dc=lbl ,dc=gov' -aproxyPassword=<password> bcsb-ldap-2.als.lbl.gov***
- **Edit pam.conf and nsswitch.conf**

Implementation on Solaris (cont'd)



- **nsswitch.conf:**

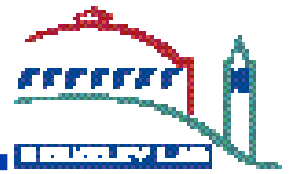
```
passwd:      files ldap
group:       files ldap
```

Or

```
passwd:          compat files
passwd_compat:   ldap
group:           files ldap
```

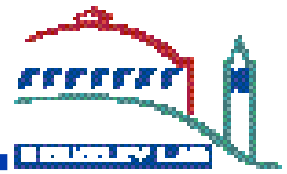
- **The former form provides (in the absence of other filtering) full access to the directory contents.**
- **The latter is the form to use for netgroup-based restrictions.**

Red Hat Linux



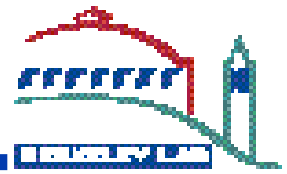
- Red Hat Linux is by far the easiest implementation we've done
- authconfig utility handles all the work of editing pam.conf, and the nsswitch.conf mods are much saner than the default Solaris nsswitch.conf

OS X



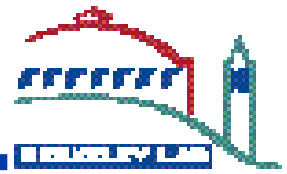
- Directory Access utility sets up access to LDAP directories
- Doesn't handle being disconnected from the network well

Tru64 Unix



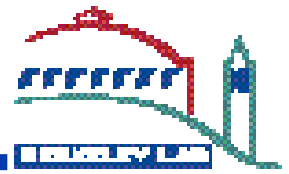
- No PAM on Tru64
- HP provides Internet Express package, which includes LDAP authentication and light nameservice functionality
- LDAPv2-only, which may cause problems with OpenLDAP > 2.1.0 (and will definitely not work out-of-the-box)
- Very, very fussy

Windows



- If predominately Windows and you want LDAP, Active Directory is (*shudder*) probably the way to go.
- AD could power everything else I've listed
- However, if Windows is in the minority, Samba can do the job

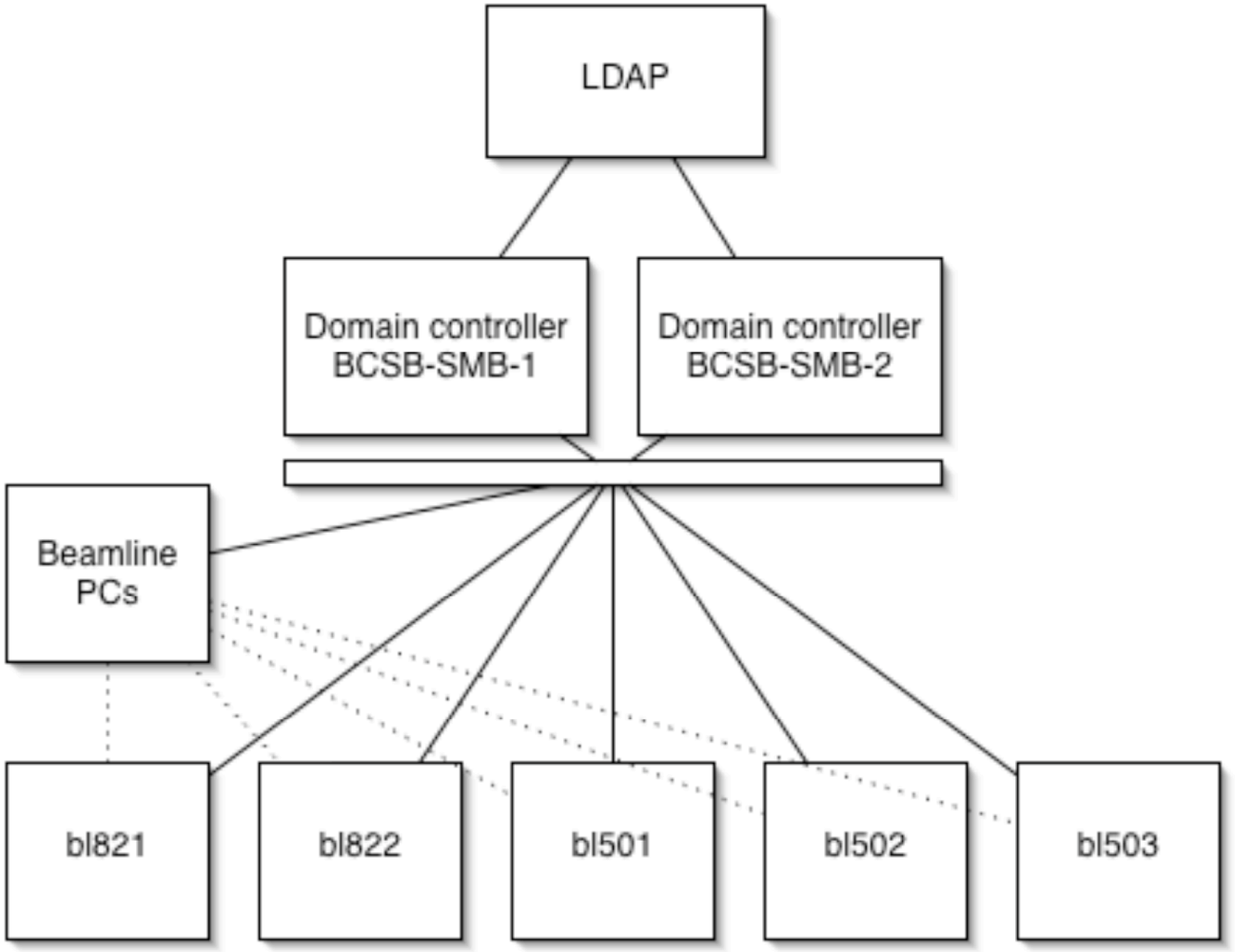
Samba



- Samba can act as Primary Domain Controller
- Samba can store SAM database in LDAP
- Samba can access LDAP for NT and LM passwords

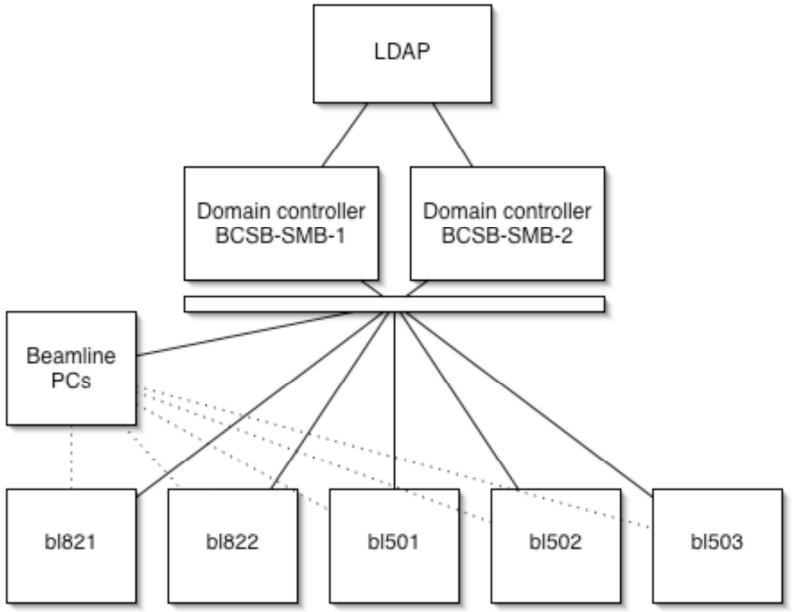
- Our solution: Samba PDC to integrate Windows machines.
- Wins include unified home directories, unified passwords, unified userbase.

Samba Implementation

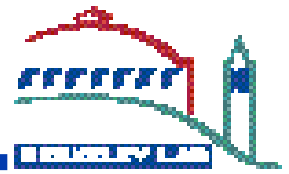


Samba Implementation

- Samba PDCs store SAM in LDAP database.
- PCs on beamlines talk both to central LDAP substrate for home directories, they also talk to individual Samba servers on beamline servers.
- Beamline servers are configured to use domain controllers as password servers.
- Result: One authenticated login to domain allows access to all Windows resources.



Apache

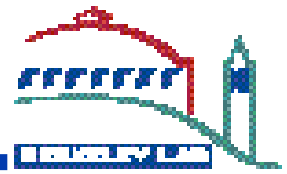


- Integration with Apache via mod_auth_ldap (as well as several mod_perl modules)
- Filter users using LDAP filters, or required particular DNs, groups, etc.
- For quick authentication against the Lab's LDAP server:

```
AuthLDAPUrl ldap://ldap.lbl.gov/ou=people,o=Lawrence Berkeley
    Laboratory,c=US
require valid-user
```

- mod_auth_ldap defaults to looking for uid; it will retrieve the DN from the LDAP server.
- BCSB dropped mod_auth_ldap to shift to Tomcat Realms

Tomcat



- **Easy to implement LDAP authentication using Tomcat (>4) authentication realms:**

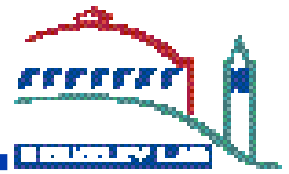
From server.xml:

```
<Realm
        className="org.apache.catalina.realm.JNDIRealm"
        connectionURL="ldap://bcsb-ldap.als.lbl.gov:389"
        contextFactory="com.sun.jndi.ldap.LdapCtxFactory"
        debug="0"
        roleBase="ou=groups,dc=als,dc=lbl,dc=gov"
        roleSubtree="false"
        userRoleName="host"
        userPattern="uid={0},ou=people,dc=als,dc=lbl,dc=gov"
        userSubtree="false"
        validate="true"/>
```

From web.xml in application:

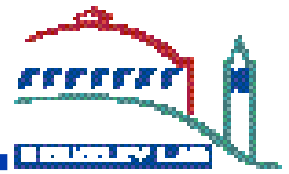
```
<auth-constraint>
    <!-- Anyone with one of the listed roles may access this area -->
    <role-name>admin</role-name>
</auth-constraint>
```

Support applications



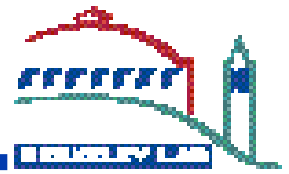
- passwd replacement - required to change both LDAP and Samba passwords together.
- useradd replacement - required to create Samba user info, as well as role-based flags
- UserAdmin - Tomcat-based user management application

Problems and Issues



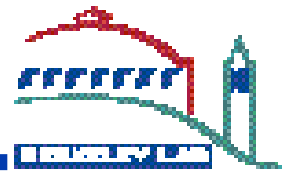
- Solaris: TLS tough to implement, new on Solaris 8
- Solaris: goofy default nsswitch.conf
- Solaris: Integration of PADL stuff tricky, caused some problems
- Solaris: tilde completion (cd ~gre<tab>) problem; seemingly fixed -- after 2 years open
- Tru64: Avoid if at all possible
- OS X: Not good for intermittently connected machines
- Linux, OS X, Tru64: Failover between servers is lacking

Overall Experience



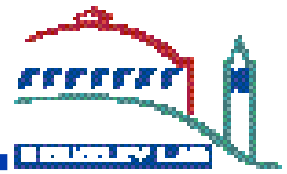
- Integrating platforms is more complex and involved than what most people suggest
- Once working, the system is a dream.
  - I never add users -- others can do that
  - I never have to change passwords, others can do that
- When something does “break,” it’s usually traceable to client flakiness or user error
- If you’ve got oddball (non-PAM) platforms, such as Tru64 or IRIX, do some serious testing; don’t accept propaganda at face value.

Overall Experience (cont'd)



- Give hierarchy design a lot of thought. It can be changed later, but you probably don't want to.
- Be sure to define *all* of your goals *before* you implement.
- Be sure you get yourself into a directory frame of mind. It will make visualization much, much easier. (I.e., don't let yourself fall into the relational-db way of thinking.)

Q&A



- ??